J. Symbolic Computation (2000) **30**, 341–356
doi:10.1006/jsco.2000.0411
Available online at http://www.idealibrary.com on IDEAL

Computing Ideals of Points

J. ABBOTT[†§], A. BIGATTI[†§], M. KREUZER[‡¶] AND L. ROBBIANO[†§]

[†] *Department of Mathematics, University of Genova, Italy*
[‡] *Department of Mathematics, University of Regensburg, Germany*

We address the problem of computing ideals of polynomials which vanish at a finite set of points. In particular we develop a modular Buchberger–Möller algorithm, best suited for the computation over $\mathbb{Q}$, and study its complexity; then we describe a variant for the computation of ideals of projective points, which uses a direct approach and a new stopping criterion. The described algorithms are implemented in CoCoA, and we report some experimental timings.

1. Introduction

The easiest geometric object in affine or projective space is a single rational point. It has no secrets, in particular its defining ideal, i.e. the set of all the polynomials which vanish at the point, is straightforward to describe. Namely, for an affine point with coordinates $(a_1, \ldots, a_n)$, the corresponding ideal is $\mathfrak{p} = (x_1 - a_1, \ldots, x_n - a_n)$; while for a projective point with coordinates $(a_0 : \ldots : a_n)$, the corresponding homogeneous ideal is $\mathfrak{B} = (a_i x_1 - a_1 x_i, \ldots, a_i x_n - a_n x_i)$ for any i such that $a_i \neq 0$. But obviously the matter becomes more complicated if we want the algebraic description of the vanishing ideal of a finite set of points. Let K be a field, and let $\mathbb{A}^n_K$ and $\mathbb{P}^n_K$ denote, respectively, the n-dimensional affine and projective spaces over K. Then let $\mathbb{X} := \{P_1, \ldots, P_s\}$ be a finite set of s rational points; that is, points having coordinates in the field K. Why are we interested in the defining ideal $I(\mathbb{X})$? And how do we compute it?

We answer the first question by listing just a few examples we know of where ideals of points have been used in different fields of mathematics.

(1) They encode fundamental properties of algebraic varieties of which they are linear sections (see, for instance, Geramita *et al.*, 1993, and the references indicated there).
(2) They are the basis for interpolation problems of numerical analysis (see Möller, 1998).
(3) They are used in coding theory (see Sakata, 1998).
(4) They have been recently introduced into the realm of statistics, where finite sets of points are called designs and fractions (see Robbiano, 1998).

As to the second question, we observe that $I(\mathbb{X}) = \bigcap_{i=1}^{s} I(P_i)$, hence there is an obvious answer: it is possible to use a well-known byproduct of Buchberger's Algorithm to compute the intersection. However, this approach is not efficient from the computational

[§] E-mail: {abbott,bigatti,robbiano}@dima.unige.it
[¶] E-mail: martin.kreuzer@mathematik.uni-regensburg.de

0747–7171/00/100341 + 16 $35.00/0

point of view and to remedy this an algorithm (called BM-algorithm) of low complexity was presented in Buchberger and Möller (1982); moreover, this algorithm has good performance (see, for instance, Marinari *et al.* (1993)). Several generalizations have since been proposed, for instance, to the cases of points with multiplicity (for instance Lakshman, 1991), and of points described by differential conditions. A variant for determining a minimal set of generators in the projective case has been studied in Cioffi (1998).

So what is left to do? The increasing demand for strong computational tools, in particular in statistics, pushed us to investigate the problem from several points of view.

First of all, previous complexity analyses referred to a model where the cost of operations in the base field is constant, but experience has shown that this is inappropriate, for instance, when the base field is $\mathbb{Q}$. So the first achievement of this paper is a *modular BM-algorithm*, a variant of the classical one, where we use a modular technique to tame the problem of coefficient growth. Our analysis of complexity given in Section 4 takes into account the varying cost of field operations, and agrees with experimental evidence.

Another important matter we address is that of points in projective space. If $\mathbb{X}$, the set of points, resides in projective space then the task of efficiently computing a Gröbner basis of $I(\mathbb{X})$ is more complicated. Let us see where the difficulties arise. Assume that the polynomial ring is $K[x_0, \ldots, x_n]$ and suppose that the hyperplane $x_0 = 0$ does not meet $\mathbb{X}$. Then it is well-known that we may proceed by considering the affine space given by $x_0 = 1$. We compute a Gröbner basis G of the ideal corresponding to the affine image of $\mathbb{X}$ with respect to some degree compatible term ordering, and then homogenize the elements of G to obtain a Gröbner basis of $I(\mathbb{X})$ with respect to an x_0-`DegRevLex` type term ordering. Now one has to use a variant of the FGLM-algorithm to obtain the Gröbner basis with respect to the desired term ordering.

But what happens if $x_0 = 0$ meets $\mathbb{X}$? The usual suggestion is to perform a suitable linear change of coordinates to avoid them meeting, then compute a Gröbner basis as described above of the transformed points, and finally reconstruct the Gröbner basis for $I(\mathbb{X})$. This approach was studied in Marinari *et al.* (1993).

Greater care is needed if the projective space is over a finite field. In this case it is clear that a finite set of points can meet every hyperplane, so that no linear change of coordinates is available to do the trick. A way around this obstacle could be to consider a field extension $K(\alpha)$ of the base field K, work in $K(\alpha)[x_0, \ldots, x_n]$, and then come back to K. This approach is clearly more demanding from the computational point of view.

These considerations suggest looking for a direct approach which avoids any change of coordinates. Indeed in Marinari *et al.* (1993) such a variant of the classical BM-algorithm was suggested. The main difficulty is that, unlike for the affine case, in the projective case ideals of points are one-dimensional, so that there is a need for an efficient stopping criterion. Unfortunately the key Lemma 12.1 in their paper is flawed and admits counterexamples (e.g. see Example 3.7). Our Section 3 is entirely devoted to the discussion of the projective case. In particular we introduce a well-behaved stopping criterion, which relies on combinatorial tools (see Theorem 3.10).

At this point the reader presumably wants to see some practical achievements of our work. All the algorithms described in this paper have been implemented in CoCoA 3.6 (see Capani *et al.*, 1998). Some experimental data based on our implementations are reported in Section 5.

In this paper we repeatedly use the notion of Gröbner basis. Needless to say this is now one of the main ingredients in Computational Algebra. The layout of the theory

of Gröbner bases can be found in the original papers by Buchberger (1965, 1970, 1985) cited in the bibliography.

2. A Modular Version of the Classical Buchberger–Möller Algorithm

Let us recall the classical version of the Buchberger–Möller Algorithm, as introduced in Buchberger and Möller (1982). We are given a set of distinct, K-rational points $\mathbb{X} = \{P_1, \ldots, P_s\}$ in n-dimensional affine space $\mathbb{A}^n$ over a field K via their coordinates, i.e. we have s tuples $P_i = (p_{i1}, \ldots, p_{in}) \in K^n$. Our goal is to compute the ideal I of all polynomials in $K[x_1, \ldots, x_n]$ which vanish at all points of $\mathbb{X}$.

Remark 2.1. Before presenting the algorithm we explain precisely what we mean by "row reduction" in a matrix. The most important aspect is that an ordering must be imposed on the columns of the matrix: the most natural is left-to-right, and by permuting the columns we may assume that this is indeed the chosen ordering. In a matrix M we say that a row whose first non-zero element occurs in column c is a **reducer** for column c; a zero row is not a reducer for any column. In Algorithm C below the matrix M is constructed so that every row is a reducer, and no column has more than one associated reducer. A row vector may then be reduced against M by repeatedly subtracting suitable multiples of the reducers in M to kill the first non-zero element in the vector. The process ends when the vector becomes zero or when there is no reducer in M for the column in which the first non-zero entry lies.

Algorithm C. (Classical Buchberger–Möller Algorithm) *Let K be a field, let $n \geq 1$, let σ be a term ordering on the power products $\mathbb{T}^n$ of $K[x_1, \ldots, x_n]$, and let $P_i = (p_{i1}, \ldots, p_{in}) \in K^n$ for $i = 1, \ldots, s$. Consider the following sequence of instructions.*

C1 *Start with empty lists $G = [\,]$, $Q = [\,]$, $S = [\,]$, a list $L = [1]$, and a matrix $M = (m_{ij})$ over K, with s columns and initially zero rows.*

C2 *If $L = [\,]$, return the pair $[G, Q]$ and stop. Otherwise, choose the power product $t = \min_\sigma(L)$, the smallest according to the ordering σ. Delete t from L.*

C3 *Compute the evaluation vector $(t(P_1), \ldots, t(P_s)) \in K^s$, and reduce it against the rows of M to obtain*

$$(v_1, \ldots, v_s) = (t(P_1), \ldots, t(P_s)) - \sum_i a_i(m_{i1}, \ldots, m_{is}) \qquad a_i \in K.$$

C4 *If $(v_1, \ldots, v_s) = (0, \ldots, 0)$ then append the polynomial $t - \sum_i a_i s_i$ to the list G, where s_i is the ith element of S. Remove from L all multiples of t. Continue with step C2.*

C5 *Otherwise $(v_1, \ldots, v_s) \neq (0, \ldots, 0)$, so add $(v_1, \ldots, v_s)$ as a new row to M, and $t - \sum_i a_i s_i$ as a new element to S. Append the power product t to Q, and add to L those elements of $\{x_1 t, \ldots, x_n t\}$ which are neither multiples of an element of L nor of $\mathrm{LT}_\sigma(G)$. Continue with step C2.*

Theorem 2.2. *Algorithm C stops after finitely many steps. It returns the reduced σ-Gröbner basis G of the ideal I of all polynomials vanishing on $\mathbb{X} = \{P_1, \ldots, P_s\}$, and the list of power products Q whose residue classes form a K-basis of $K[x_1, \ldots, x_n]/I$.*

PROOF. See Marinari *et al.* (1993). □

With only a slight change, Algorithm C can also yield the separators f_i of the points P_i, i.e. polynomials such that $f_i(P_i) = 1$ and $f_i(P_j) = 0$ whenever $i \neq j$. To do this replace step C2 by the following:

C2 bis *If $L = [\,]$, reduce M to an identity matrix by row operations mimicking these row operations on the elements of S, and then return the triple $[G, Q, S]$ and stop. Otherwise, choose the power product $t = \min_\sigma(L)$ and delete it from L.*

COROLLARY 2.3. *Algorithm C with step C2 bis additionally computes a set of separators for the points P_i; these separators are reduced with respect to the Gröbner basis G.*

PROOF. Observe that throughout Algorithm C the matrix M and list S are related as follows: row i of M is just $(s_i(P_1), s_i(P_2), \ldots, s_i(P_s))$ where s_i is the ith element of the list S. Moreover, the last time step C2 bis is entered M is a non-singular $s \times s$ matrix as the points are distinct. The linear combinations of the rows of M yielding the identity matrix clearly also convert the elements of S into separators. □

Conceptually, Algorithm C can be split into two principal tasks: determining the quotient basis Q (and implicitly $\mathrm{LT}(G)$), and then computing G from $\mathrm{LT}(G)$ by linear algebra.

Algorithm C works well when the field K is finite but is less well-suited to the case $K = \mathbb{Q}$ where growth of the entries of M becomes a problem. For this reason we present a modular version of the algorithm: a commonly used paradigm for avoiding intermediate expression swell. Before describing the algorithm we recall some useful facts.

The Chinese Remainder Algorithm (CRA) converts a collection of residue-modulus pairs $\{r_i \bmod p_i : i = 1, 2, \ldots, e\}$ into a single number R such that $R \equiv r_i \bmod p_i$ for all i provided the moduli p_i are pairwise coprime. R is determined uniquely modulo $\prod_{i=1}^{e} p_i$. In particular, if the r_i are known to be the residues of some integer bounded in absolute value by B, then the integer value can be determined trivially from R provided $\prod_{i=1}^{e} p_i > 2B$.

A rational number can also be recovered from its image modulo a sufficiently large modulus. There are two ways to do this. If we can compute a multiple of its denominator then we can clear the denominator and find the integral numerator as above. Otherwise, if no (reasonable) multiple of the denominator is known then we can use the rational recovery algorithm of Wang *et al.* (1982) which was later refined in Collins and Encarnación (1995). Applying either of these approaches when the modulus is not big enough will produce some "meaningless" rational number (or a warning that no suitable rational exists).

The question of how many primes to use can be resolved in two ways: determine *a priori* bounds on the numerators and denominators and then deduce how many primes to use, or speculatively recover the rationals after including information from each prime and check whether the result is correct. The second approach is useful when no good bound is available and the result can easily be checked for correctness; this is the method we use.

REMARK 2.4. The main loop of the algorithm has repeatedly to select a new prime p for the modular computation, but the prime must be suitable. The conditions p has to satisfy are:

(1) p must be different from the other primes chosen so far;
(2) the coordinates of the points $P_1, \ldots, P_s$ should have reductions modulo p, i.e. the prime p must not divide any of their denominators;
(3) the reductions $\overline{P}_1, \ldots, \overline{P}_s \in \mathbb{F}_p^n$ should remain pairwise distinct points.

ALGORITHM M. (MODULAR VERSION OF THE BUCHBERGER–MÖLLER ALGORITHM) *Let $n \geq 1$, let σ be a term ordering on $\mathbb{T}^n$, and let $P_i = (p_{i1}, \ldots, p_{in}) \in \mathbb{Q}^n$ for $i = 1, \ldots, s$. Consider the following sequence of instructions.*

M1 *Let $Q = [\,]$, $G = [\,]$ and $G_m = [\,]$ be empty lists, and let $m = 1$.*
M2 *Pick a new prime number p (see Remark 2.4).*
M3 *Apply Algorithm C (the classical Buchberger–Möller Algorithm) to $K = \mathbb{F}_p$ and the set of points $\{\overline{P}_1, \ldots, \overline{P}_s\}$, where $\overline{P}_i = (\bar{p}_{i1}, \ldots, \bar{p}_{in}) \in \mathbb{F}_p^n$ is the reduction of P_i. Denote the result by $[G_p, Q_p]$.*
M4 *Compare Q_p with Q (see Remark 2.5). If Q is* better *than Q_p then continue with step M2. If Q_p is* better *than Q then replace Q by Q_p, set m equal to p, put G_p in G_m, then continue with step M2.*
M5 *At this point we have $Q_p = Q$. Replace m by mp. Update G_m by combining it with G_p using CRA. Perform rational recovery on G_m with modulus m, and call the result G'.*
M6 *If $G' \neq G$ then replace G by G' and continue with step M2.*
M7 *Now $G' = G$. Check whether all polynomials in G vanish at all points of the set $\{P_1, \ldots, P_s\}$. If they do not, go back to step M2. Otherwise return $[G, Q]$ and stop.*

REMARK 2.5. The list Q constructed by Algorithm C is generated in increasing order with respect to σ, and always contains exactly s power products at the end of the algorithm. If we have two such lists $Q = [q_1, q_2, \ldots]$ and $Q' = [q'_1, q'_2, \ldots]$, we say that Q is **better** than Q' if for some index j we have $q_i = q'_i$ for $i = 1, 2, \ldots, j-1$ and $q_j <_\sigma q'_j$. The idea behind this definition is that the only way for the computation modulo p to go wrong is to obtain a zero vector after the reduction in step C3 when the same computation over $K = \mathbb{Q}$ would not have found a zero vector at that point, and in this case the corresponding power product t is missing from Q_p. We justify this idea in Lemma 2.6.

LEMMA 2.6. *Let $P_1, \ldots, P_s \in \mathbb{Q}^n$ be distinct points. Let G be the Gröbner basis produced by applying Algorithm C over the field $\mathbb{Q}$ to these points, and let D be the determinant of the matrix M just before the algorithm terminated. Let p be a prime which does not divide any denominator appearing in the points, and let G_p be the Gröbner basis produced by applying Algorithm C over the field $\mathbb{F}_p$ to the modular reductions of the points. Then $G \equiv G_p \bmod p$ if and only if $D \not\equiv 0 \bmod p$.*

PROOF. First we claim that p cannot divide the (minimal) denominator of D. Consider the computation over $\mathbb{Q}$. The matrix M built up during Algorithm C is just a triangularization of the matrix E whose rows are the evaluation vectors of the quotient basis; moreover, the reduction to triangular form is achieved via a unimodular matrix. Thus $\det(M) = \pm \det(E)$, and by clearing denominators in the rows of E we see that p does

not divide the (minimal) denominator of $\det(E)$, thus proving the claim. Note that in the special case where $P_1, \ldots, P_s \in \mathbb{Z}^n$ we have $D \in \mathbb{Z}$, and our claim is trivial.

Next we show that $\mathrm{LT}(G) = \mathrm{LT}(G_p)$ if and only if $D \not\equiv 0 \bmod p$. Now $\mathrm{LT}(G) = \mathrm{LT}(G_p)$ if and only if $Q = Q_p$ where Q are Q_p are the respective quotient bases since Q uniquely determines $\mathrm{LT}(G)$ and vice versa; similarly for Q_p and $\mathrm{LT}(G_p)$. But Q and Q_p differ if and only if the reduction in step C3 produces a zero vector modulo p where a non-zero vector was obtained over $\mathbb{Q}$, and this happens if and only if there is a linear dependency modulo p amongst the rows of E, i.e. $D \equiv 0 \bmod p$.

If $D \not\equiv 0 \bmod p$ then by Cramer's rule we see that the solution $\underline{x}$ to any linear system $M\underline{x} = \underline{b}$ has least common denominator not divisible by p whenever the least common denominator of $\underline{b}$ is not divisible by p. This implies that p does not divide the least common denominator of any element of G (nor of any separator).

Finally we show that $\mathrm{LT}(G) = \mathrm{LT}(G_p)$ if and only if $G \equiv G_p \bmod p$. Since the elements of G and G_p are monic, the reverse implication is trivial. So suppose that $\mathrm{LT}(G) = \mathrm{LT}(G_p)$. Then necessarily $Q = Q_p$, and G is obtained from $\mathrm{LT}(G)$ by representing the evaluation vectors of each element of $\mathrm{LT}(G)$ as a linear combination of the rows of E. G_p is obtained analogously. Now we have just seen that each element of G can be reduced modulo p since we must have $D \not\equiv 0 \bmod p$. This gives a representation modulo p of the evaluation vector of some element of $\mathrm{LT}(G)$ as a linear combination of the rows of E_p (the matrix whose rows are the evaluation vectors modulo p of the quotient basis), but E_p is invertible modulo p, so the representation is unique, and thus coincides with an element of G_p. □

THEOREM 2.7. *Algorithm M stops after finitely many steps. It returns the reduced σ-Gröbner basis G of the vanishing ideal I of $\mathbb{X} = \{P_1, \ldots, P_s\}$ and a list of power products Q whose residue classes form a K-basis of $K[x_1, \ldots, x_n]/I$.*

PROOF. We shall prove termination and correctness together. Let $[G^*, Q^*]$ denote the result Algorithm C would have produced given the same input, and let M^* be the final value of the matrix used there (but not returned).

In step M3 we can have $Q_p \neq Q^*$ only finitely many times since by Lemma 2.6 it implies that p divides $\det(M^*)$. Furthermore, once a prime has been found for which $Q_p = Q^*$ then in Algorithm M we will have $Q = Q^*$ ever after.

Algorithm M cannot return with $Q^* \neq Q$. For suppose we manage to reach step M7 with $Q^* \neq Q$. Then G contains an element, g, with leading term outside $\mathrm{LT}(G^*)$, hence g itself lies outside the ideal generated by G^*, and consequently the verification will fail. So we must eventually pick a prime p with $Q_p = Q^*$.

Now assume that $Q^* = Q$. Any prime for which $Q_p \neq Q^*$ will be discarded by step M4; so assume also that $Q_p = Q^*$. By Lemma 2.6 we must also have $G \equiv G_p \bmod p$. Hence once we have $Q = Q^*$, then G_m in step M5 will be the reduction modulo m of G, and so for sufficiently large m the rational recovery will yield G, at which point the verification in step M7 will succeed. □

COROLLARY 2.8. *Algorithm M can easily be adapted to compute the separators as well (using the variant of Algorithm C).*

PROOF. Note that the final verification in step M7 must check the values at each point of both the Gröbner basis elements and the separators.

Recall that the separators are obtained by representing the vectors $(1,0,0,0,\ldots)$, $(0,1,0,0,\ldots)$, etc., as linear combinations of the evaluation vectors of the quotient basis. Now it is easy to extend the proof of Theorem 2.7. □

3. Computing Ideals of Points in Projective Space

Here we describe the direct approach, as promised in the introduction. In the affine case, the ideal of a set of points is zero-dimensional, and hence the algorithm works on a finite set of power products. In the projective case, the ideal of a set of points is one-dimensional, so we do not have an "obvious" way to deduce when the computation of the Gröbner basis is completed.

The first step towards a stopping criterion comes from the theory of Hilbert functions of sets of points in $\mathbb{P}^n$. We recall some definitions and results (see Mora and Robbiano, 1993, and Geramita *et al.*, 1993). To avoid tedious repetition we shall assume throughout this section that the points $P_1,\ldots,P_s$ in $\mathbb{P}^n_K$ are distinct and K-rational.

Definition 3.1. Let $A = K[x_0,\ldots,x_n]$, let I be a homogeneous ideal in A. The **Hilbert function** of A/I is defined as $H_{A/I} : \mathbb{N} \longrightarrow \mathbb{N}$ with $H_{A/I}(d) = \dim_K(A/I)_d$.

Theorem 3.2. (Hilbert Function) *Let $\mathbb{X} = \{P_1,\ldots,P_s\}$ be a set of points in $\mathbb{P}^n_K$, and let $I \subseteq A = K[x_0,\ldots,x_n]$ be the homogeneous vanishing ideal of $\mathbb{X}$.*

(1) There exists $a \geq -1$ such that $H_{A/I}(d) < H_{A/I}(d+1)$ for all $d \leq a$, and $H_{A/I}(d) = s$ for all $d > a$.
In particular, since $H_{A/I}(0) = 1$, we have $H_{A/I}(d) = s$ for all $d \geq s-1$.

(2) For any term ordering σ on A we have $H_{A/I} = H_{A/\operatorname{LT}_\sigma(I)}$.

Definition 3.3. For $\mathbb{X} = \{P_1,\ldots,P_s\}$ a set of points in $\mathbb{P}^n_K$, we define $a_{\mathbb{X}}$ to be the integer a of Theorem 3.2.

In the following proposition we use a term ordering σ of x_0-`DegRevLex` type: this means that σ is degree compatible and for two power products t, t' of the same degree we have $t >_\sigma t'$ whenever $x_0 | t'$ but $x_0 \not| t$. Consequently, for such a term ordering, if $x_0 | \operatorname{LT}_\sigma(f)$, then $x_0 | f$.

Proposition 3.4. (Simple Stopping Criterion) *Let $\mathbb{X} = \{P_1,\ldots,P_s\} \subseteq \mathbb{P}^n_K$ be a set of points with homogeneous vanishing ideal $I \subseteq A = K[x_0,\ldots,x_n]$. Suppose that the hyperplane $H = \mathcal{Z}(x_0)$ contains no point of $\mathbb{X}$, and that the term ordering σ is of x_0-`DegRevLex` type. Then the leading term ideal $\operatorname{LT}_\sigma(I)$ is generated in degrees $\leq a_{\mathbb{X}} + 2$.*

Proof. First note that for each $d > 0$ the map $(A/\operatorname{LT}_\sigma(I))_{a_{\mathbb{X}}+d} \to (A/\operatorname{LT}_\sigma(I))_{a_{\mathbb{X}}+d+1}$ defined by multiplication by x_0 is an isomorphism. This follows from the fact that the two vector spaces have the same dimension, the hypothesis that x_0 is not a zero-divisor modulo I, and the assumed type of σ.

Now we show that for $d > 2$ we have $\operatorname{LT}_\sigma(I)_{a_{\mathbb{X}}+d} \subseteq A_1 \operatorname{LT}_\sigma(I)_{a_{\mathbb{X}}+d-1}$; and hence by iteration that $\operatorname{LT}_\sigma(I)_{a_{\mathbb{X}}+d} \subseteq A_{d-2} \operatorname{LT}_\sigma(I)_{a_{\mathbb{X}}+2}$.

Let $d > 2$ and $f \in \operatorname{LT}_\sigma(I)_{a_{\mathbb{X}}+d}$. We can write $f = \sum_{i=0}^n x_i f_i$ with $f_0,\ldots,f_n \in A_{a_{\mathbb{X}}+d-1}$. Using one of the isomorphisms above, we can express $f_i = x_0 g_i + h_i$ where each $g_i \in A_{a_{\mathbb{X}}+d-2}$ and each $h_i \in \operatorname{LT}_\sigma(I)_{a_{\mathbb{X}}+d-1}$. Thus $f = x_0 \sum_{i=0}^n x_i g_i + \sum_{i=0}^n x_i h_i$,

and this implies $x_0 \sum_{i=0}^{n} x_i g_i \in \mathrm{LT}_\sigma(I)_{a_\mathbb{X}+d}$, and therefore $\sum_{i=0}^{n} x_i g_i \in \mathrm{LT}_\sigma(I)_{a_\mathbb{X}+d-1}$. Altogether, this proves that $f \in A_1 \,\mathrm{LT}_\sigma(I)_{a_\mathbb{X}+d-1}$. □

Our next examples show that this stopping criterion fails in general.

Example 3.5. Consider this set of points $\mathbb{X} = \{(0,2,5),(0,1,2),(1,3,1),(4,3,4),(2,5,4),(1,4,4)\}$ in $\mathbb{P}^2_\mathbb{Q}$. The values of the associated Hilbert function are 1, 3, 6, 6, ..., hence $a_\mathbb{X} = 1$. There are separators for all six points in each degree $d \geq 2$, and I is generated in degree ≤ 3. However, for $\sigma = \texttt{DegRevLex}$, the leading term ideal of I is $(xyz, xz^2, yz^2, z^3, xy^3)$. Thus there is one Gröbner basis element in degree $4 > a_\mathbb{X} + 2$. This failure depends on the fact that x_0 divides zero modulo I. Indeed two points are on the line $x_0 = 0$.

Example 3.6. For this set of five points $\mathbb{X} = \{(4,1,1),(5,2,-1),(85,4,1),(455,8,-1),(4369,16,1)\}$ in $\mathbb{P}^2_\mathbb{Q}$ the values of the associated Hilbert function are $1,3,5,5,\ldots$, thus $a_\mathbb{X} = 1$ again. However using $\sigma = \texttt{DegLex}$ we obtain $\mathrm{LT}_\sigma(I) = (x^2, xz^2, xy^2, y^5)$ showing that there is an element in the reduced σ-Gröbner basis of I of degree 5. This failure depends on the fact that σ is not of x_0-`DegRevLex` type.

The example below shows that during the execution of the algorithm the fact that the Hilbert function attains its maximum in some degree does not imply that the Gröbner basis is complete (or nearly complete). In particular it shows that the key Lemma 12.1 of Marinari *et al.* (1993) is incorrect.

Example 3.7. Consider the set of 15 points $\mathbb{X} = \{(n,m,1)\}_{n,m\in\{-1,0,1,2\}} \setminus \{(2,2,1)\}$ in $\mathbb{P}^2_\mathbb{Q}$. The values of the associated Hilbert function are $1,3,6,10,13,15,15,\ldots$, so we have $H(5) = 15$. Now, there is no generator of $\mathrm{LT}_{\texttt{DegRevLex}}(I)$ in degree 5, and the generators of degree < 5 do generate an ideal of dimension 1 (projective dimension 0). But there is another generator of the ideal in degree 6. In fact, the ideal I is minimally generated by

$$\{y^4 - 2y^3z - y^2z^2 + 2yz^3,\quad x^4 - 2x^3z - x^2z^2 + 2xz^3,\quad x^3y^3 - x^3yz^2 - xy^3z^2 + xyz^4\}.$$

Definition 3.8. Let $\mathbb{T}^{n+1}_d$ be the set of power products in $K[x_0,\ldots,x_n]$ of degree d, and let $Q \subseteq \mathbb{T}^{n+1}_d$. Two power products t, t' are **connected in** Q if there is a sequence of power products $t_0, t_1, \ldots, t_r \in Q$ with $t_0 = t$ and $t_r = t'$ such that for each $i = 1,\ldots,r$ there exist $\alpha, \beta \in \{0,\ldots,n\}$ satisfying $t_i = t_{i-1} \cdot x_\alpha / x_\beta$. The intuitive meaning of this notion is that one can pass from one t_i to the next by replacing one indeterminate by another.

We define the **connected components in** Q in the obvious way; they are clearly disjoint. The connected component of a power product $t \notin Q$ is empty.

For example, in $\mathbb{Q}[x,y,z]$, let $Q := \{x^3, x^2z, xz^2, yz^2, y^3\}$. In Q we have that x^3 is connected to yz^2: a connecting sequence is x^3, x^2z, xz^2, yz^2. The power product y^3 is connected only to itself.

Lemma 3.9. (Disconnection Lemma) *Let J be a monomial ideal in $A := K[x_0,\ldots,x_n]$ and $Q_\delta := \mathbb{T}^{n+1}_\delta \setminus J_\delta$ for every $\delta \in \mathbb{N}$. Suppose that there exists $d \in \mathbb{N}$ such that:*

(1) $H_{A/J}(d) = s$, i.e. $|Q_d| = s$;

(2) for each $i \in \{0,1,\ldots,n\}$, every power product in the connected component of x_i^d in Q_d is divisible by x_i;

(3) $H_{A/J}(\delta) \geq s$ *for all* $\delta \geq d$.

Then $H_{A/J}(\delta) = s$ *for all* $\delta \geq d$.

PROOF. Let C_i be the connected component of x_i^d in Q_d. It is empty only if $x_i^d \in J$. By hypothesis (2) these components are distinct, hence disjoint, therefore $\sum_{i=0}^n |C_i| \leq s$.

First we prove that (1) and (2) imply $Q_\delta \subseteq \bigcup_{i=0}^n x_i^{\delta-d} C_i$ for all $\delta \gg d$: let J' be the ideal generated by $J_d \cup C_0 \cup \cdots \cup C_n$. Obviously J' contains all x_i^d, hence $J'_\delta = \mathbb{T}_\delta^{n+1}$ for all $\delta \gg d$, in particular for all $\delta \geq (n+1)d$.

Now we prove that for each i we have $\{x_0, \ldots, x_n\} \cdot C_i \subseteq J \cup x_i C_i$. From the definition of the C_i, for any $t \in C_i$ and any x_j we have that $t' = t \cdot x_j / x_i$ is either in J or in C_i. Whence $x_j \cdot t = x_i \cdot t' \in J \cup x_i C_i$.

Therefore $J'_{d+1} = J_{d+1} \cup (\bigcup_i \{x_0, \ldots, x_n\} C_i)$ equals $J_{d+1} \cup (\bigcup_i x_i C_i)$, and by induction it easily follows that $J'_\delta = J_\delta \cup (\bigcup_i x_i^{\delta-d} C_i)$ for all $\delta \geq d$. This means that Q_δ, the complement of J_δ, is wholly contained in $\bigcup_i x_i^{\delta-d} C_i$ for all $\delta \gg d$.

Now we prove that $|Q_\delta| = s$ for all $\delta \geq d$: let $\bar{\delta} \gg d$. From what we have proved and hypothesis (3) it follows that $\sum_{i=0}^n |C_i| \geq |Q_{\bar{\delta}}| \geq s$ and therefore $\sum_{i=0}^n |C_i| = s$, which implies $Q_d = \bigcup_{i=0}^n C_i$. Thus $J'_d = \mathbb{T}_d^{n+1}$ and then $J'_\delta = \mathbb{T}_\delta^{n+1}$ for all $\delta \geq d$, hence $Q_\delta \subseteq \bigcup_{i=0}^n x_i^{\delta-d} C_i$ for all $\delta \geq d$.

This and hypothesis (3) imply $|Q_\delta| = s$ for all $\delta \geq d$, which is exactly what we wanted to prove. □

THEOREM 3.10. (STOPPING CRITERION) *Let* $\mathbb{X} = \{P_1, \ldots, P_s\}$ *be a set of points in* $\mathbb{P}_K^n$, *let* I *be the homogeneous vanishing ideal of* $\mathbb{X}$, *and let* σ *be a term ordering on* $A = K[x_0, \ldots, x_n]$. *Assume that there exists* d *such that*

(1) $H_{A/\operatorname{LT}_\sigma(I)}(d)$ *equals the number of points in* $\mathbb{X}$;
(2) for all i, *every power product in the connected component of* x_i^d *in* $\mathbb{T}_d^{n+1} \setminus \operatorname{LT}_\sigma(I)$ *is divisible by* x_i.

Then the elements of the reduced σ*-Gröbner basis of* I *have degrees* $\leq d$.

PROOF. Let s be the number of points in $\mathbb{X}$ and let J be the ideal generated by the leading power products of the elements of the reduced σ-Gröbner basis of I of degree $\leq d$. We want to prove $J = \operatorname{LT}_\sigma(I)$.

Applying Theorem 3.2 to our hypothesis, we have

$$H_{A/J}(\delta) \geq H_{A/\operatorname{LT}_\sigma(I)}(\delta) = H_{A/I}(\delta) = s \quad \text{for all} \quad \delta \geq d.$$

Hence, by Lemma 3.9, we have $H_{A/J}(\delta) = s = H_{A/I}(\delta)$ for all $\delta \geq d$ and then $H_{A/J}(\delta) = H_{A/\operatorname{LT}_\sigma(I)}(\delta)$ for all $\delta \in \mathbb{N}$. This and the fact that $J \subseteq \operatorname{LT}_\sigma(I)$ imply $J = \operatorname{LT}_\sigma(I)$. □

Based on this theorem we define the function `StoppingCriterion`(s, Q) which returns "TRUE" only if Q contains s elements of maximal degree d and the connected components of every x_i^d in Q satisfy condition (2).

ALGORITHM CP. (PROJECTIVE VERSION OF THE CLASSICAL BUCHBERGER–MÖLLER ALGORITHM) *Let* K *be a field, let* $n \geq 1$, *let* σ *be a term ordering on the monoid* $\mathbb{T}^{n+1}$ *of power products of* $K[x_0, \ldots, x_n]$, *and let* $P_i = (p_{i0} : \ldots : p_{in}) \in \mathbb{P}^n(K)$ *for* $i = 1, \ldots, s$. *Consider the following sequence of instructions.*

CP1 *Start with empty lists* $G = [\,]$, $S = [\,]$, *lists* $Q = [1]$ *and* $L = [1]$, *a matrix* $M = (m_{ij})$ *over* K *with* s *columns and initially zero rows, and with* $d = 0$.

CP2 *If the function* `StoppingCriterion`(s, Q) *defined above yields TRUE then return the pair* $[G, Q]$ *and stop. Otherwise increase* d *by one, reset* M *to be a matrix over* K *with zero rows and* s *columns, reset* $S = [\,]$, *and let* L *be the list of all power products of degree* d *which are not multiples of a leading power product (w.r.t.* σ*) of an element of* G.

CP3 *If* L *is empty, go to step CP2; otherwise choose the power product* $t = \min_\sigma(L)$ *and remove it from* L.

CP4 *Compute the evaluation vector* $(t(P_1), \ldots, t(P_s)) \in K^s$, *and reduce it against the rows of* M, *to obtain*

$$(v_1, \ldots, v_s) = (t(P_1), \ldots, t(P_s)) - \sum_i a_i(m_{i1}, \ldots, m_{is}) \qquad a_i \in K.$$

CP5 *If* $(v_1, \ldots, v_s) = (0, \ldots, 0)$ *then append the polynomial* $t - \sum_i a_i s_i$ *to the list* G, *where* s_i *is the* i*th element of the list* S. *Continue with step CP3.*

CP6 *Otherwise* $(v_1, \ldots, v_s) \neq (0, \ldots, 0)$ *so add* $(v_1, \ldots, v_s)$ *as a new row to* M *and* $t - \sum_i a_i s_i$ *as a new element to* S. *Append the power product* t *to* Q. *Continue with step CP3.*

REMARK 3.11. The value Q returned by Algorithm CP is normally of little utility; we return it only to simplify the enunciation of Algorithm MP (below).

THEOREM 3.12. *Algorithm CP stops after finitely many steps and returns the reduced* σ*-Gröbner basis* G *of the homogeneous vanishing ideal* I *of* $\mathbb{X} = \{P_1, \ldots, P_s\}$.

PROOF. First we show finiteness of the algorithm: in particular, we prove that the algorithm never goes beyond degree s. From the construction we see that each time step CP2 is entered the number of power products in the list Q is exactly $H_{A/I}(d)$. From Theorem 3.2 part 1 we have that $H_{A/I}(d) = s$ for all degrees $d \geq s - 1$. Let us consider the case $d = s$. We have seen that there are precisely s power products of degree s in Q, that is condition (1) of the stopping criterion. Also condition (2) is satisfied: to connect x_i^s to some power product not divisible by x_i needs at least s steps, i.e. a sequence containing at least $s + 1$ power products. Hence the stopping criterion applies.

To show correctness we prove that $\mathrm{LT}_\sigma(I)$ is generated by $\mathrm{LT}_\sigma(G)$. Consider the end of step CP2. The new list L contains all power products of degree d which are not in the ideal generated by $\mathrm{LT}_\sigma(G)$. Since, by construction, all elements of G vanish on $\mathbb{X}$, we have $G \subseteq I$, hence $\mathrm{LT}_\sigma(G) \subseteq \mathrm{LT}_\sigma(I)$. We have already remarked that upon arriving at step CP2 the number of power products in the list Q of degree d is just $H_{A/I}(d)$. Therefore $H_{A/\mathrm{LT}_\sigma(G)} = H_{A/I}$. This implies that $H_{A/\mathrm{LT}_\sigma(G)} = H_{A/\mathrm{LT}_\sigma(I)}$, which, together with $\mathrm{LT}_\sigma(G) \subseteq \mathrm{LT}_\sigma(I)$, yields the desired conclusion. □

The following example shows that the stopping criterion is not always optimal.

EXAMPLE 3.13. Consider the set of points $\mathbb{X} = \{(1,1,1), (0,1,1), (0,1,0), (1,1,0)\}$ in $\mathbb{P}^2_{\mathbb{Q}}$. The leading term ideal w.r.t. $\sigma =$ `DegRevLex` of its homogeneous vanishing ideal I is $\mathrm{LT}_\sigma(I) = (yz, x^2)$. In degree 2, the stopping criterion is not satisfied, because z^2 is connected to xy via $[z^2, xz, xy]$. In degree 3, the stopping criterion is satisfied, but there is no new σ-Gröbner basis element in that degree.

Definition 3.14. Let $A = K[x_0, \ldots, x_n]$, let I be a homogeneous ideal in A, and let $\mathbb{X} = \{P_1, \ldots, P_s\} \subseteq \mathbb{P}^n(K)$ be a set of points. A homogeneous polynomial $f \in A$ is called a **separator** of P_i, if $f(P_i) \neq 0$ and $f(P_j) = 0$ for $j \neq i$. A separator of minimal degree is also called a **minimal separator**.

Remark 3.15. As earlier for Algorithm C we may easily adapt Algorithm CP to compute separators. The initial value for d in step CP1 should be -1 to handle correctly the case of just a single input point. In step CP6 we must additionally use the newly added row to bring M into reduced row echelon form. Then in step CP2 once the stopping criterion has been fulfilled we obtain the separators as a subset of the list S: if row i in M contains exactly one non-zero entry, in column j say, then s_i is a separator for the point P_j. To obtain minimal separators, in step CP2 we must check for rows of M containing a single non-zero element and take the corresponding elements of S prior to resetting M and S; if there are several separators for the same point, we retain only the one of lowest degree.

Algorithm MP. (Modular Version of the Projective Buchberger–Möller Algorithm) *This is identical to Algorithm M except that it calls Algorithm CP in step MP3 where Algorithm C was called in step M3.*

Theorem 3.16. *Algorithm MP stops after finitely many steps. It returns the reduced σ-Gröbner basis G of the vanishing ideal I of $\mathbb{X} = \{P_1, \ldots, P_s\}$.*

Proof. This is entirely analogous to the proof of Theorem 2.7. □

Corollary 3.17. *Algorithm MP can be adapted to compute the separators as well (using the variant of Algorithm CP described in Remark* 3.15*).*

Proof. The adaptation is straightforward, as for Algorithm M, except for two points which do not arise in the affine case. The projective separators are defined only up to multiplication by a non-zero field element. So that the separators can be recovered from their modular images we must eliminate this ambiguity: we do this by scaling the separators so that they evaluate to 1 at the given representatives of their own points.

To obtain minimal separators we must also be wary of "bad" separators: it is possible that the minimal separator modulo p of some point has lower degree than the minimal separator over $\mathbb{Q}$. In such a case the "false modular image" must be discarded as it would otherwise inhibit recovery of the separators over $\mathbb{Q}$. The simplest approach is to compare the degrees of the separators for the "current prime" with the degrees of the separators built up by Chinese Remaindering: if some separator modulo p has degree lower than the corresponding lifted separator then discard all information from that prime; otherwise if some separator modulo p has degree higher than the corresponding lifted separator then discard all lifted values and start anew from the current prime. Other more efficient approaches exist but they are more complicated. □

4. Complexity Issues

In this section we determine a worst case complexity for Algorithm M to compute both the separators of the points and the Gröbner basis for their ideal. We also determine the

complexity for "generic" points with the term ordering `DegRevLex`; by "generic" we mean random points whose behaviour is as that of true generic points.

As in the paper by Marinari *et al.* (1993) the parameters we shall use to express the complexity are:

s the number of points;
n the number of variables (i.e. the dimension of the ambient affine space);
g the number of elements in the Gröbner basis.

Our Algorithm C is the same as Algorithm 1 presented in Section 5 of that paper. We recall the result of their analysis which showed that the complexity of applying Algorithm C with $K = \mathbb{F}_p$ is $O(s^2(g+s)(\log p)^2 + s^2n^2)$ where the cost of an arithmetic operation in $\mathbb{F}_p$ is $O((\log p)^2)$. Our result is encapsulated in the following theorem.

THEOREM 4.1. *For the case $K = \mathbb{Q}$, where all points have integral coordinates with largest magnitude X, and X satisfies $\log(X) \in O(\log s)$, i.e. X is bounded by some polynomial in s, then Algorithm M has worst case bit complexity*

$$O(s^4(g+s)(\log s)^2 + s^4n^2).$$

Furthermore, for "generic" points with the `DegRevLex` *ordering, Algorithm M has bit complexity*

$$O(s^3d(g+s)(\log s)^2 + s^3dn^2)$$

where d is such that $\binom{n+d-1}{d} = s + g$; this is better than the worst case complexity by a factor of s/d.

PROOF. We start by estimating how many bad primes we could encounter. Consider the $s \times s$ matrix, M, whose columns are indexed by the points in $\mathbb{X}$ and whose rows are indexed by the power products in the basis of the quotient R/I; the entries being the value of the power product at the point. By definition this matrix is non-singular; moreover, by Lemma 2.6 a prime is "bad" if and only if it divides $\det(M)$. Now we estimate the greatest possible value for this determinant using Hadamard's bound on the rows. In any row corresponding to a power product of total degree d no entry can exceed X^d in absolute value. Using $d_1, d_2, \ldots, d_s$ to denote the degrees of the power products for the rows in our matrix, we see that Hadamard's bound does not exceed $X^D\sqrt{s^s}$ where $D = d_1 + \cdots + d_s$. Since our quotient basis contains every factor of every element of the basis, we deduce that $D \leq 1 + 2 + \cdots + s = s(s+1)/2$. This worst case is realized with generic points and the `Lex` ordering; in contrast, generic points with the `DegRevLex` ordering will typically give a much smaller exponent of $D = sd$ where d is the highest degree of a Gröbner basis element. Here we note that the logarithm of Hadamard's bound in the worst case is: $O(s^2 \log X + s \log s)$.

For any positive integer r the product of the first r primes clearly exceeds $r!$, and this we can estimate using Stirling's asymptotic approximation: $\log(r!) = r \log r - r + o(r)$. With the given hypothesis on $\log X$ we see that the number of bad primes is $O(s^2)$; in practice it appears that bad primes are typically fairly uncommon.

Next we find out how many good primes we need to be able to reconstruct the answer. To do this we must estimate the sizes of every coefficient of every separator and Gröbner basis element. Again we use the matrix M from above, and shall apply Cramer's rule

for obtaining the solutions to a linear system since our coefficients effectively arise from solving linear systems. It is immediate that the determinant of M is a common denominator for both separators and Gröbner basis elements. Hadamard's bound applied to the numerators given by Cramer's rule shows that no numerator exceeds $X^{s+1}|\det(M)|$. From this we see that the logarithm of the product of the good primes may have to become as large as $O(s^2 \log X + s \log s)$ in the worst case. Thus the total number of primes, both good and bad, which could be necessary is $O(s^2)$—experiments suggest that random points in $\mathbb{A}^2$ with coordinates in the range $[-2s^2, 2s^2]$ achieve this complexity. The situation is far more favourable for generic points with `DegRevLex` ordering: the same reasoning shows that the total number of primes needed is only $O(sd)$.

So we call Algorithm C up to $O(s^2)$ times at a total cost of $O(s^4(g+s)(\log s)^2 + s^4n^2)$ where the factor of $(\log s)^2$ allows for the fact that an arithmetic operation modulo an s-bit prime costs $O((\log s)^2)$. Following the same line of reasoning we can also deduce that for generic points using `DegRevLex` ordering the total cost of the calls to Algorithm C is $O(s^3d(g+s)(\log s)^2 + s^3dn^2)$.

We comment that the cost of the calls to Algorithm C is at least that of Chinese Remaindering and verification combined. Combining $O(s^2)$ modular values (each of size $O(\log s)$) into a single value using iterative Chinese Remaindering costs $O(s^3(\log s)^2)$, and we must do this for $s(s+g)$ coefficients, giving a total cost of $O(s^4(g+s)(\log s)^2)$. Verifying the answer by evaluation costs no more than $O(s^4g(\log s)^2)$: the s power products can each be evaluated at the s points once only at a total cost of $O(s^4 \log s)$, and then each polynomial in the Gröbner basis has no more than $s+1$ terms and can be evaluated in time $O(s^4(\log s)^2)$. It can similarly be shown that, in the case of generic points with the `DegRevLex` ordering, the overall complexity is the same as that of the calls to Algorithm C. □

Now we determine the complexity for Algorithm MP (for the field $\mathbb{Q}$). We continue to use the same parameters but observe that now n is one greater than the dimension of the ambient projective space. First we establish the complexity of Algorithm CP over the field $K = \mathbb{F}_p$: the algorithm never goes beyond degree s (see the proof of Theorem 3.12), and in each degree there are at most s power products not giving rise to Gröbner basis elements, so step CP4 is executed at most $s^2 + g$ times. Step CP4 costs at most $s(n+s)$ operations in K, and this dominates other costs giving a total complexity of

$$O(s(n+s)(g+s^2)(\log p)^2 + s^2n^2 \log(ns))$$

assuming that each arithmetic operation in $\mathbb{F}_p$ costs $O((\log p)^2)$; the second summand in the complexity accounts for the cost of generation of the power products to be considered. Our result for Algorithm MP is encapsulated in the following theorem.

THEOREM 4.2. *For the case $K = \mathbb{Q}$, where all points have integral coordinates with largest magnitude X, and satisfies $\log(X) \in O(\log s)$, i.e. X is bounded by some polynomial in s, then Algorithm MP has worst case bit complexity*

$$O(s^3(n+s)(g+s^2)(\log s)^2 + s^4n^2 \log(ns)).$$

Furthermore, for "generic" points with the `DegRevLex` *ordering, Algorithm MP has bit complexity*

$$O(s^2d(n+s)(g+sd)(\log s)^2 + s^3dn^2 \log(ns))$$

where d is such that $\binom{n+d-1}{d} = s + g$; this is better than the worst case complexity by a factor of about s/d.

PROOF. This proof is largely analogous to that of Theorem 4.1. In place of the single matrix M considered in that proof we must consider several matrices $M_1, M_2, \ldots, M_d$ where the rows of M_r are indexed by certain power products of degree r. Since we never need to go beyond degree s we have that $d \leq s$. Each of these matrices is, by definition, of full rank, and a prime is "bad" if and only if at least one M_r is no longer of full rank modulo p; i.e. p divides the determinant of every maximal minor of at least one M_r. Hadamard's bound for the determinant of a maximal minor of M_r gives $X^r\sqrt{s^s}$. Thus all bad primes must divide a number of size not exceeding $\log \prod_{r=1}^{s}\left(X^r\sqrt{s^s}\right) \in O(s^2 \log X + s^2 \log s)$. Whence there can be at most s^2 bad primes (cf. proof of Theorem 4.1).

The maximum number of "good" primes we could ever need remains $O(s^2)$ since we never need to go beyond degree s: we have to consider solving linear systems corresponding to the matrices $M_1, M_2, \ldots, M_d$, and arguing as in the proof of Theorem 4.1, Cramer's Rule and Hadamard's bound prove that the logarithm of the product of the good primes need never exceed $O(s^2 \log X + s \log s)$ in the worst case.

The argument that the cost of the main loop dominates remains valid for this case. Since we may have to execute the main loop up to $O(s^2)$ times the overall complexity is just s^2 times the complexity of the call to Algorithm CP.

We note that the case of generic points with the `DegRevLex` ordering permits a much more favourable analysis. All elements of the Gröbner basis have degree $d-1$ or d (where $d \in O(\log s / \log n)$). We shall show that a prime p is "bad" if and only if at least one of M_{d-2}, M_{d-1} or M_d is not of full rank modulo p; as in the proof of Theorem 4.1 we can now deduce that the total number of primes to be tried is $O(sd)$, and thus the complexity in this case is just sd times the complexity of a call to Algorithm CP, and this latter decreases slightly since we only go as far as degree d instead of all the way to degree s.

Now we prove that to characterize "bad" primes it suffices to consider only the three matrices M_{d-2}, M_{d-1} and M_d. Let p be a "bad" prime for which the matrix M_r does not have full rank for some $r < d-2$. This implies that there is a polynomial f of degree r which vanishes (modulo p) at every point of $\mathbb{X}$. Now consider the polynomial $x^{d-r-2}f$; this has degree $d-2$ and clearly vanishes (modulo p) at every point of $\mathbb{X}$. But, since the points are generic, the rows of M_{d-2} are indexed by *all* power products of degree $d-2$, hence there is a non-trivial linear combination of the rows of M_{d-2}; i.e. the rank of M_{d-2} is not full. □

5. Implementation Issues and Timings

Algorithm M as described admits a number of implementation refinements. For instance, at step M4 the comparison of the lists Q and Q_p can be effected inside Algorithm C as the list is generated; Algorithm C can then terminate prematurely if a worse Q is being generated.

The rational recovery and stability checks at steps M5 and M6 need only monitor a single coefficient most of the time, though eventually the full answer will be needed. It is also possible to try both rational recovery techniques until one of them yields a stable answer; this is what our implementation does.

The verification in step M7 need not actually evaluate any polynomials: we represent a polynomial with rational coefficients as f/D where D is the least common denominator,

Table 1.

DegRevLex	$\mathbb{A}^{10}$	$\mathbb{A}^{20}$	$\mathbb{A}^{40}$	$\mathbb{P}^{9}$	$\mathbb{P}^{19}$	$\mathbb{P}^{39}$	$\mathbb{P}^{39}$
40 small	0.3 s	0.7 s	3.0 s	0.4 s	0.8 s	3.1 s	14 digits
40 medium	1.7 s	2.8 s	8.1 s	2.9 s	5.2 s	12.0 s	45 digits
40 large	3.2 s	5.7 s	14.0 s	7.7 s	12.0 s	23.0 s	123 digits
40 extralarge	7.0 s	12.9 s	26.0 s	11.8 s	22.0 s	47.0 s	264 digits
80 small	3.7 s	5.6 s	14.0 s	2.7 s	5.6 s	17.0 s	39 digits
80 medium	22.0 s	24.0 s	66.0 s	42.0 s	46.0 s	96.0 s	157 digits
80 large	53.0 s	48.0 s	109.0 s	89.0 s	120.0 s	180.0 s	349 digits
80 extralarge	116.0 s	110.0 s	250.0 s	196.0 s	180.0 s	360.0 s	640 digits

Table 2.

Lex	$\mathbb{A}^{10}$	$\mathbb{A}^{20}$	$\mathbb{A}^{40}$
40 small	0.1 s	0.2 s	0.3 s
40 medium	0.4 s	0.6 s	0.9 s
40 large	3.3 s	4.0 s	5.4 s
40 extralarge	29.7 s	38.2 s	59.0 s
80 small	0.5 s	0.6 s	0.9 s
80 medium	2.5 s	2.6 s	4.3 s
80 large	24.0 s	27.0 s	36.0 s
80 extralarge	490.0 s	590.0 s	720.0 s

and f has integer coefficients; we know the polynomials are correct modulo m, so it suffices to verify that $|f(P_i)| < m$ for each point P_i, and this can be done quickly and easily (e.g. by bounding each term separately and summing these). This cheap verification test may require $O(s)$ further iterations of the main loop before recognizing that the answer has been found; this extra cost does not affect the complexity, and in practice it works well. An analogous cheap verification works for the separators too.

Tables 1 and 2 give timings for the computation of ideals of distinct random points, computed on a Digital Alpha, 192 Mb RAM, 433 MHz, compiled with `gcc -O2`. The first column indicates the number and the size of the points: *small* means that the coordinates are integers between -1 and 1, *medium* between -9 and 9, *large* between -99 and 99, and *extralarge* between -9999 and 9999. The other columns give the dimension and type of space (affine or projective). The last column in Table 1 gives the average size of the coefficients in the basis of points in $\mathbb{P}^{39}_{\mathbb{Q}}$. This shows very clearly how the timings over $\mathbb{Q}$ depend on the size of the coordinates.

All these timings are given for computations over $\mathbb{Q}$. The computation of these examples over finite fields is performed by comparison in negligible time; for example the ideal of 80 points in $\mathbb{P}^{39}$ w.r.t. `DegRevLex` over $\mathbb{Z}/(32003)$ is computed in 2.8 seconds.

Note the apparently surprising behaviour when comparing the orderings `DegRevLex` and `Lex` on the smaller examples. This is because we used random points from a relatively small range rather than generic points, so the `Lex` basis comes out "smaller than normal" (smaller coefficients and lower maximal degree of the leading terms of the basis). The "extralarge" rows show the expected behaviour.

Acknowledgement

This work was partially supported by the Consiglio Nazionale delle Ricerche (CNR).

References

Buchberger, B. (1965). An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal. Ph.D. Thesis, University of Innsbruck (Austria). (In German).

Buchberger, B. (1970). An algorithmical criterion for the solvability of algebraic systems of equations. *Aequationes Math.*, **4**, 374–383.

Buchberger, B. (1985). In Bose, N. K. ed., *Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory*, chapter 6, pp. 184–232. D. Reidel Publishing Co.

Buchberger, B., Möller, H. M. (1982). The construction of multivariate polynomials with preassigned zeros. In Calmet, J. ed., *Proceedings of the European Computer Algebra Conference (EUROCAM'82) (Marseille, France)*, LNCS **144**, pp. 24–31. Springer.

Capani, A., Niesi, G., Robbiano, L. (1998). *CoCoA, A System for Doing Computations in Commutative Algebra*. Available via anonymous ftp from `cocoa.dima.unige.it`, 3.6 edition.

Cioffi, F. (1998). Minimally generating ideals of points in polynomial time using linear algebra. *Ric. Mat., XLVII fasc. 2*.

Collins, G. E., Encarnación, M. J. (1995). Efficient rational number reconstruction. *J. Symb. Comput.*, **20**, 287–298.

Geramita, A. V., Kreuzer, M., Robbiano, L. (1993). Cayley-Bacharach schemes and their canonical modules. *Trans. Am. Math. Soc.*, **339**, 163–189.

Lakshman, Y. (1991). A single exponential bound on the complexity of computing Gröbner bases of zero dimensional ideals. In Mora, T., Traverso, C. eds, *MEGA-90 (Castiglioncello/Italy)*, volume 94 of *Progress in Mathematics*, pp. 227–234. Boston, Birkhäuser.

Marinari, M., Moeller, H., Mora, T. (1993). Gröbner bases of ideals defined by functionals with an application to ideals of projective points. *Appl. Alg. Eng., Commun. Comput.*, **4**, 103–145.

Mora, T., Robbiano, L. (1993). Points in affine and projective spaces. In Eisenbud, D., Robbiano, L. eds, *Computational Algebraic Geometry and Commutative Algebra, Cortona-91*, volume 34 of *Symposia Mathematica*, pp. 106–150. Cambridge University Press.

Möller, M. (1998). Gröbner bases and numerical analysis. In Buchberger, B., Winkler, F. eds, *Gröbner Bases and Applications (Proceedings of the Conference on 33 Years of Gröbner Bases)*, volume 251 of *London Mathematical Society Lecture Notes Series*, pp. 159–178. Cambridge University Press.

Robbiano, L. (1998). Gröbner bases and statistic. In Buchberger, B., Winkler, F. eds, *Gröbner Bases and Applications (Proceedings of the Conference on 33 Years of Gröbner Bases)*, volume 251 of *London Mathematical Society Lecture Notes Series*, pp. 179–204. Cambridge University Press.

Sakata, S. (1998). Gröbner bases and coding theory. In Buchberger, B., Winkler, F. eds, *Gröbner Bases and Applications (Proceedings of the Conference 33 Years of Gröbner Bases)*, volume 251 of *London Mathematical Society Lecture Notes Series*, pp. 205–220. Cambridge University Press.

Wang, P. S., Guy, M. J. T., Davenport, J. H. (1982). p-adic reconstruction of rational numbers. *SIGSAM Bull. (ACM Special Interest Group on Symbolic and Algebraic Manipulation)*, **16**, 2–3.

Originally Received 31 January 1999
Accepted 6 February 1999